Global numerical approach to nonlinear discrete time control

Luc Jaulin, Eric Walter

Global Numerical Approach to Nonlinear Discrete-Time Control

Luc Jaulin and Eric Walter

Abstract—Interval analysis is used to characterize the set of all input sequences with a given length that drive a nonlinear discrete-time state-space system from a given initial state to a given set of terminal states. No requirement other than computability (i.e., ability to be evaluated by a finite algorithm) is put on the nature of the state equations. The method is based on an algorithm for set inversion and approximates the solution set in a guaranteed way.

Index Terms— Characterization of sets, interval analysis, nonlinear control, set inversion.

NOMENCLATURE

Plain lower case letters (x):	Scalars.
Plain capital letters (X):	Scalar intervals.
Bold lower case letters ($\mathbf{x}$):	Vectors.
Bold capital letters ($\mathbf{X}$):	Vector intervals, or boxes.
Outlined capital letters ($\mathbb{V}$):	Sets that are not necessarily intervals or boxes.

I. INTRODUCTION

Consider an nth-order discrete-time system described by the state equation

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{1}$$

where $\mathbf{u}(k)$ is the input vector, $\mathbf{x}(k)$ is the state at time k, $\mathbf{x}_0$ is some known initial state, and $\mathbf{f}$ is a known nonlinear vector function. Various approaches have been proposed to control such a system. One may, for instance, look for a feedback law that transforms it into a linear system [1] and then apply some linear technique such as pole placement. The results obtained are often local and rely on analyticity conditions. One may also search for the sequence of inputs that is best in the sense of some optimality criterion (see, e.g., [2]). Here also, the results obtained are most often local, because of the nonconvexity of the criterion. At best, one may expect to get one of the possible solutions, even if there are several of them.

Manuscript received June 1, 1994; revised April 27, 1995 and July 10, 1996.

L. Jaulin is with the Equipe Signaux, Systèmes et Applications, Faculté des Sciences, Université d'Angers, 49045 Angers Cedex 01, France.

E. Walter is with the Laboratoire des Signaux et Systèmes, CNRS-SUPELEC, 91192 Gif-sur-Yvette Cedex, France (e-mail: walter@lss.supelec.fr).

Publisher Item Identifier S 0018-9286(97)03592-7.

The problem considered here is the *guaranteed* characterization of the set $\mathbf{V}$ of *all* input sequences $\mathbf{v} = [\mathbf{u}^T(0), \cdots, \mathbf{u}^T(m-1)]^T$ of given length m that drive the state of the system from $\mathbf{x}_0$ to some given terminal set $\mathbf{X}$. Methods for obtaining one such control sequence for a discrete-time linear system with a closed and convex target $\mathbf{X}_t$ are presented in [3], but the guaranteed characterization of the set of all possible solutions in a nonlinear context seems to be addressed here for the first time. A control sequence $\mathbf{v}$ of length m will be said to be *feasible* if

$$\mathbf{g}(\mathbf{v}) \triangleq \mathbf{f}(\mathbf{f}(\cdots(\mathbf{f}(\mathbf{x}(0), \mathbf{u}(0)), \cdots), \mathbf{u}(m-2)), \mathbf{u}(m-1)) \in \mathbf{X}_t. \tag{2}$$

Thus, the set of all feasible input sequences of length m is given by

$$\mathbf{V} = \mathbf{g}^{-1}(\mathbf{X}_t) \tag{3}$$

where $\mathbf{g}^{-1}$ is the reciprocal function of $\mathbf{g}$ in a set-theoretic sense. If $\mathbf{f}$ is polynomial in $\mathbf{x}$ and $\mathbf{u}$, $\mathbf{g}$ is polynomial in $\mathbf{v}$. When $\mathbf{X}_t$ is a singleton, solving (3) for $\mathbf{V}$ then amounts to finding all solutions of a set of polynomial equations in several unknowns. Many methods have been proposed for that purpose. Global continuation methods are numerical and based on the notion of homotopy paths (see, e.g., [4]). Methods based on elimination theory and computer algebra transform the set of polynomial equations into a simpler one (often triangular) with the same solutions (see, e.g., [5]). Interval variants of the Newton method can also be used to approximate the set of all solutions numerically, but in a guaranteed way [6].

Contrary to those mentioned above, the method to be presented is not limited to the case where $\mathbf{X}_t$ is a singleton. It does not require $\mathbf{f}$ to have any specific characteristic (such as analyticity, continuity or differentiability) other than being computable by a finite algorithm, which permits one to take into account nonlinearities such as saturations or thresholds. Although the method is numerical, it provides global and guaranteed results.

The algorithm set inverter via interval analysis (SIVIA), introduced in the context of bounded-error estimation [7], [8] and applied to the characterization of stability domains [9], will be used to approximate $\mathbf{V}$ by solving the set-inversion problem (3). The minimum knowledge about interval analysis required to understand the procedure is recalled in Section II. The algorithm for set inversion is described in Section III. Examples are treated in Section IV.

II. Interval Analysis

Interval analysis [10] is a fundamental numerical tool for proving properties of sets, solving sets of nonlinear equations or inequalities, and optimizing nonconvex criteria in a guaranteed way. A *box*, or *vector interval* $\mathbf{X}$ of R^n is the cartesian product of n real intervals X_i

$$\mathbf{X} = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] = X_1 \times \cdots \times X_n. \tag{4}$$

Denote the set of all boxes of $\mathbb{R}^n$ by $\mathbb{IR}^n$. Let $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^p$ be a vector function; the set-valued function $\mathbf{F}\colon \mathbb{IR}^n \to \mathbb{IR}^p$ is a (convergent) *inclusion function* of $\mathbf{f}$ if, for any box $\mathbf{X}$ of $\mathbb{IR}^n$, it satisfies the two following conditions:

$$\mathbf{f}(\mathbf{X}) \subset \mathbf{F}(\mathbf{X}) \tag{5}$$

$$w(\mathbf{X}) \to 0 \Rightarrow w(\mathbf{F}(\mathbf{X})) \to 0 \tag{6}$$

where $w(\mathbf{X})$ is the *width* of the box $\mathbf{X}$, i.e., the length of its largest side(s). Note that $\mathbf{f}(\mathbf{X})$ is usually not a box, contrary to $\mathbf{F}(\mathbf{X})$, which is a box by definition. The calculation of an inclusion function for any computable function (i.e., given by a finite algorithm) is usually very simple [10] and routinely performed by commercially available languages such as PASCAL XSC [11].

Example 1: An inclusion function for

$$\mathbf{f}\colon \begin{pmatrix} u \\ x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_1 * \cos(x_1 * x_2) + u \\ 3x_1^2 - \sin(u * x_2) \end{pmatrix} \tag{7}$$

is

$$\mathbf{F}\colon \begin{pmatrix} U \\ X_1 \\ X_2 \end{pmatrix} \to \begin{pmatrix} X_1 * \cos(X_1 * X_2) + U \\ 3X_1^2 - \sin(U * X_2) \end{pmatrix}. \tag{8}$$

If, for instance, $\mathbf{X} = [0, 1] \times [0, \pi/3]$ and $U = [-2, 1]$, then the box $\mathbf{F}(\mathbf{X}, U)$ is computed as follows:

$$\begin{aligned} \mathbf{F}(\mathbf{X}, U) &= \begin{pmatrix} [0,1] * \cos([0,1] * [0, \pi/3]) + [-2, 1] \\ 3 * [0,1]^2 - \sin([-2,1] * [0, \pi/3]) \end{pmatrix} \\ &= \begin{pmatrix} [0,1] * \cos([0, \pi/3]) + [-2, 1] \\ 3 * [0,1] - \sin([-2\pi/3, \pi/3]) \end{pmatrix} \\ &= \begin{pmatrix} [0,1] * [0.5, 1] + [-2, 1] \\ [0, 3] - [-1, \sqrt{3}/2] \end{pmatrix} \\ &= \begin{pmatrix} [0,1] + [-2, 1] \\ [-\sqrt{3}/2, 4] \end{pmatrix} = \begin{pmatrix} [-2, 2] \\ [-\sqrt{3}/2, 4] \end{pmatrix}. \end{aligned} \tag{9}$$

Note that the operators $+, *, -$ and functions cos, sin, and $(\cdot)^2$ in (8) are interval counterparts of those in (7). Replacing each elementary operator and function by its interval counterpart is but one method to obtain an inclusion function, usually far from being the most effective. The resulting inclusion function is called *natural inclusion function*.

III. Algorithm for Set Inversion

The algorithm SIVIA proceeds directly from the notion of the inclusion function. Its aim is to build two *subpavings* (i.e., sets of nonoverlapping boxes), $\mathbf{V}^-$ and $\mathbf{V}^+$, so as to bracket the set $\mathbf{V}$ defined by (3) between an inner and an outer approximation

$$\mathbf{V}^- \subset \mathbf{V} \subset \mathbf{V}^+. \tag{10}$$

SIVIA uses a *stack* of boxes, i.e., a first-in-first-out set of boxes (think of a stack of plates), on which three operations are possible, namely *stacking* (putting a box on the top), *unstacking* (removing the top box), and testing the stack for emptiness. The current box $\mathbf{V}$ is initially taken equal to the prior box of interest (i.e., some possibly very large box in the space of the input sequences of length m, in which the search is going to be performed) and then split whenever no conclusion is reached, unless its width is smaller than some given required accuracy ε_r.

SIVIA can be summarized as follows.

Inputs:

Prior box of interest: $\mathbf{V}_0$.
Inclusion function for $\mathbf{g}(\cdot)$: $\mathbf{G}(\cdot)$.
Width of the smallest box allowed to be bisected: ε_r.

Initialization:

Stack $:= \emptyset, \mathbf{V}^- := \emptyset, \mathbf{V}^+ := \emptyset, \mathbf{V} = \mathbf{V}_0$.

Iteration:

Step 1) If $\mathbf{G}(\mathbf{V}) \subset \mathbf{X}_t$, then $\mathbf{V}^- := \mathbf{V}^- \cup \mathbf{V}$; $\mathbf{V}^+ := \mathbf{V}^+ \cup \mathbf{V}$, go to Step 4).
Step 2) If $\mathbf{G}(\mathbf{V}) \cap \mathbf{X}_t = \emptyset$, then go to Step 4).
Step 3) If $w(\mathbf{V}) \le \varepsilon_r$, then $\mathbf{V}^+ := \mathbf{V}^+ \cup \mathbf{V}$, else bisect $\mathbf{V}$ along a symmetry plane perpendicular to one of its largest edges and stack the two resulting boxes.
Step 4) If the stack is not empty, then unstack into $\mathbf{V}$ and go to Step 1).
Step 5) Output $\mathbf{V}^-$ and $\mathbf{V}^+$; end. (11)

If the condition of Step 1) is satisfied, all $\mathbf{v}$ in $\mathbf{V}$ are feasible, and $\mathbf{V}$ is appended to $\mathbf{V}^-$ and $\mathbf{V}^+$. If the condition of Step 2) is satisfied, no $\mathbf{v}$ in $\mathbf{V}$ is feasible and $\mathbf{V}$ is discarded. If none of the conditions of Steps 1) and 2) is satisfied and if $\mathbf{V}$ is large enough, then it is split into subboxes to be considered again at a later stage. Checking the conditions of Steps 1) and 2) is trivial since $\mathbf{G}(\mathbf{V})$ is a box. SIVIA is a finite algorithm that terminates in less than $(w(\mathbf{V}_0)/\varepsilon_r + 1)^{\dim \mathbf{v}}$ iterations [7], which corresponds to the degenerate situation where all boxes remain indeterminate. The complexity of the algorithm was also studied in [7]. The main limitations of SIVIA lie in the exponential increase of the computing time and number of boxes to be considered with the dimension of the box $\mathbf{V}$, here the length m of the input sequence times the dimension of the input vector $\mathbf{u}$. Even for a very large dim $\mathbf{v}$, the maximum size of the stack remains extraordinarily small. For instance, if dim $\mathbf{v} = 100, w(\mathbf{V}_0) = 10^4$, and $\varepsilon_r = 10^{-10}$, then it can be proved [7] that card(Stack) < 4600. Convergence conditions were given in [8]; under continuity conditions, i.e., the assumption that a small variation of $\mathbf{X}_t$ results in a small variation of $\mathbf{V}$, (10) defines a neighborhood of $\mathbf{V}$ in the set of compacts with a diameter that tends to zero with ε_r. The characterization of $\mathbf{V}$ can then be made as precise as desired, at the cost of increasing the computation. An algorithm similar to SIVIA was developed independently [12], with the stack replaced by a queue, which increases the memory requirements quite considerably.

Note that if $\mathbf{V}^+$ turns out to be empty, the set-inversion problem is guaranteed to have no solution.

IV. Application to Control

SIVIA applies directly to the global and guaranteed characterization of the set of all control sequences that drive the system from $\mathbf{x}_0$ to $\mathbf{X}_t$. Sequences of increasing length m can be studied until a nonempty solution set is obtained. Two cases can be considered, depending on how $\mathbf{X}_t$ is defined.

The first case is when $\mathbf{X}_t$ is defined by a set of equalities $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, which may for instance correspond to the largest linearizable manifold [13] or to the equation $\mathbf{x} = \mathbf{0}$. The set of control sequences to be characterized is then given by $\mathbf{V} = (\mathbf{h} \circ \mathbf{g})^{-1}(\mathbf{0})$, and SIVIA will produce a subpaving $\mathbf{V}^+$ guaranteed to contain it. If $\mathbf{V}^+$ is empty, the algorithm has proved that no feasible control sequence of length m exists. Except for pathological cases [8], $\mathbf{V}$ is of zero measure, so $\mathbf{V}^-$ remains empty even when $\mathbf{V}^+$ is not. However, the distance to $\mathbf{X}_t$ of the terminal state resulting from a control sequence in $\mathbf{V}^+$ can be made as small as desired by decreasing ε_r.

The second case is when $\mathbf{X}_t$ is defined by a set of inequalities $\mathbf{h}(\mathbf{x}) \leq \mathbf{0}$, which may for instance correspond to a region where it is possible to switch to a local approach. The set of control sequences to be characterized $\mathbf{V} = (\mathbf{h} \circ \mathbf{g})^{-1}((\mathbb{R}^-)^n)$, if not empty, is generally not of zero measure, so it becomes possible to obtain a nonempty $\mathbf{V}^-$.

Two examples will now be considered. The first corresponds to the case where $\mathbf{X}_t$ is the singleton $\{\mathbf{0}\}$ and the second to the case where $\mathbf{X}_t$ is a box. In both problems, the input is scalar, and the length of the input sequence is two. The number dim $\mathbf{v}$ of control variables is therefore two, which facilitates the presentation of the results. The method obviously applies to vector inputs and longer input sequences, but the dimension of the problems that can be handled is limited by the exponential complexity of the algorithm with respect to dim $\mathbf{v}$. All examples have been treated with a program written in PASCAL and run on a Compaq 386/33.

Example 2: Consider the discrete-time state-space model

$$\begin{cases} x_1(k+1) = x_1(k) * \cos(x_1(k) * x_2(k)) + u(k) \\ x_2(k+1) = 3x_1^2(k) - \sin(u(k) * x_2(k)) \end{cases}$$

$$\mathbf{x}(0) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \tag{12}$$

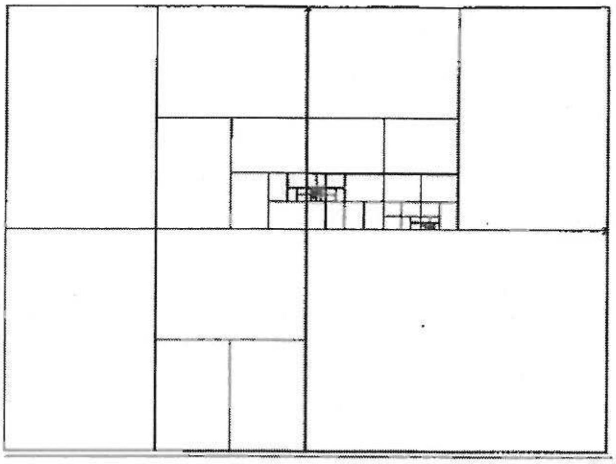

Fig. 1. Boxes eliminated by SIVIA for Example 2. The five boxes remaining in $\mathbf{V}^+$ are too small to be seen. The frame corresponds to the prior box of interest $[-1, 1]^2$ in the $(u(0), u(1))$ space.

Driving it back to the origin in two steps amounts to finding a vector

$$\mathbf{v} = \begin{pmatrix} u(0) \\ u(1) \end{pmatrix} \tag{13}$$

that satisfies $\mathbf{g}(\mathbf{v}) = \mathbf{0}$, where $\mathbf{g}(\mathbf{v})$ is computed by the pseudo-code:

```
x1(0) := 1; x2(0) := 2;
For k := 0 to 1 do
    begin
        x1(k + 1) := x1(k) * cos(x1(k) * x2(k)) + u(k);
        x2(k + 1) := 3x1^2(k) - sin(u(k) * x2(k));
    end;                                                    (14)
```

with

$$\mathbf{g}(\mathbf{v}) = \mathbf{g}(u(0), u(1)) := \begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix}. \tag{15}$$

An inclusion function $\mathbf{G}(\mathbf{V})$ for $\mathbf{g}(\mathbf{v})$ is thus given by the pseudo-code:

```
X1(0) := [1, 1]; X2(0) := [2, 2];
For k := 0 to 1 do
    begin
        X1(k + 1) := X1(k) * cos(X1(k) * X2(k)) + U(k);
        X2(k + 1) := 3X1^2(k) - sin(U(k) * X2(k));
    end;                                                    (16)
```

with

$$\mathbf{G}(\mathbf{V}) = \mathbf{G}(U(0), U(1)) := \begin{pmatrix} X_1(2) \\ X_2(2) \end{pmatrix}. \tag{17}$$

For a required accuracy of $\varepsilon_r = 10^{-4}$ and a prior domain of interest for the controls given by $\mathbf{V}_0 = [-1, 1]^2$, in 2 s, SIVIA produces the paving presented in Fig. 1 and generates an outer subpaving $\mathbf{V}^+$, that consists of five boxes. These five boxes form two sets of adjacent boxes, which can be enclosed in two boxes

$$\begin{aligned} \mathbf{V}_a &= [0.0268, 0.0270] \times [0.1600, 0.1603] \\ \mathbf{V}_b &= [0.4160, 0.4162] \times [0.0000, 0.0001]. \end{aligned} \tag{18}$$

$\mathbf{V}^+$ therefore satisfies

$$\mathbf{V}^+ \subset \mathbf{V}_a \cup \mathbf{V}_b. \tag{19}$$

As $\mathbf{X}_t$ is a singleton, no inner subpaving $\mathbf{V}^-$ can be obtained. Two quite distinct control strategies can therefore be considered. For instance

$$\hat{\mathbf{v}}_a = (0.0269, 0.1601)^T \quad \text{and} \quad \hat{\mathbf{v}}_b = (0.4161, 0.0000)^T \tag{20}$$

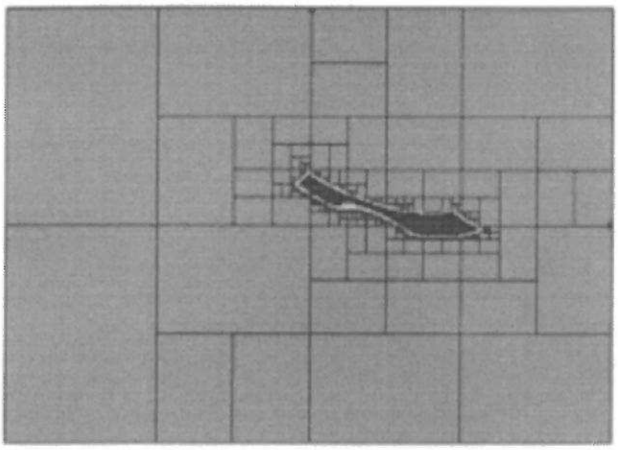

Fig. 2. Paving generated by SIVIA for Example 3. The frame is as in Fig. 1.

respectively generate the state sequences

$$\begin{aligned} \mathbf{x}_a(0) &= (1,2)^T \\ \mathbf{x}_a(1) &= (-0.3892, 2.946)^T \\ \mathbf{x}_a(2) &= (-0.000\,03, 0.0001)^T \end{aligned}$$

and (21)

$$\begin{aligned} \mathbf{x}_b(0) &= (1,2)^T \\ \mathbf{x}_b(1) &= (-0.000\,05, 2.2606)^T \\ \mathbf{x}_b(2) &= (-0.000\,05, 0.000\,00)^T. \end{aligned}$$

Example 3: Consider the same system, but assume now that we only want to drive it in two steps into the box $[-0.12, 0.12]^2$. This amounts to characterizing the set

$$\mathbf{V} = \mathbf{g}^{-1}([-0.12, 0.12]^2). \tag{22}$$

For $\varepsilon_r = 0.01$ and the same prior domain of interest for $\mathbf{v}$ as in Example 2, in 8 s, SIVIA brackets $\mathbf{V}$ between two subpavings as illustrated by Fig. 2. Boxes in dark grey belong to $\mathbf{V}^-$ and have thus been proved to belong to $\mathbf{V}$. Those in light grey have been eliminated. The uncertainty layer is in white. The complexity of the problem increases exponentially with the dimension of the accumulation set of the paving [7], which is one in this example instead of zero in Example 2. This explains why the computing time is larger than in Example 2, although ε_r is larger. Any $\mathbf{v} \in \mathbf{V}^-$ is guaranteed to send the state into $\mathbf{X}_t$.

Note that if $\mathbf{X}_t = [-0.05, 0.05]^2$, for the same required accuracy ε_r and prior domain of interest for the control, SIVIA numerically proves the nonconnexity of $\mathbf{V}$.

V. Conclusion

By taking advantage of the guaranteed nature of the numerical results provided by interval analysis, it is possible to solve the problem of computing all sequences of controls driving a deterministic nonlinear discrete-time state-space system from a given initial state to a given desired set of terminal states. To the best of our knowledge, no other guaranteed method is available for that purpose. Taking additional inequality constraints on the state or input into account would be particularly simple.

Acknowledgment

The authors wish to thank the anonymous reviewers for their help in the preparation of this paper.

References

[1] S. Monaco and D. Normand-Cyrot, "The immersion under feedback of a multidimensional discrete-time nonlinear system into a linear system," *Int. J. Contr.*, vol. 38, no. 1, pp. 245–261, 1983.

[2] M. S. Mahmoud and M. G. Singh, *Discrete Systems, Analysis, Control and Optimization*. Berlin: Springer-Verlag, 1984.

[3] R. P. Van Til and W. E. Schmitendorf, "Constrained controllability of discrete time systems," *Int. J. Contr.*, vol. 43, no. 3, pp. 941–956, 1986.

[4] C. B. Garcia and W. I. Zangwill, "Finding all solutions to polynomial systems and other systems of equations," *Math. Programming*, vol. 16, pp. 159–176, 1979.

[5] B. Buchberger, "Theoretical basis for the reduction of polynomials to canonical forms," *SIGSAM Bull.*, vol. 39, pp. 19–29, 1976.

[6] E. R. Hansen, *Global Optimization Using Interval Analysis*. New York: Marcel Dekker, 1992.

[7] L. Jaulin and E. Walter, "Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis," *Math. Comput. Simul.*, vol. 35, pp. 123–137, 1993.

[8] ——, "Set inversion via interval analysis for nonlinear bounded-error estimation," *Automatica*, vol. 29, no. 4, pp. 1053–1064, 1993.

[9] E. Walter and L. Jaulin, "Guaranteed characterization of stability domains via set inversion," *IEEE Trans. Automat. Contr.*, vol. 39, no. 4, pp. 886–889, 1994.

[10] R. E. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM, 1979.

[11] R. Klate, U. Kulisch, M. Neaga, D. Ratz, and C. Ullrich, *Pascal XSC, Language Reference with Examples*. Heidelberg: Springer-Verlag, 1992.

[12] R. E. Moore, "Parameter sets for bounded-error data," *Math. Comput. Simul.*, vol. 34, pp. 113–119, 1992.

[13] R. Marino, "On the largest feedback linearizable subsystem," *Syst. Contr. Lett.*, vol. 6, pp. 345–351, 1986.